



## Patrolling by Robots Equipped with Visibility

Jurek Czyzowicz, Evangelos Kranakis, Dominik Pajak, Najmeh Taleb

### ► To cite this version:

Jurek Czyzowicz, Evangelos Kranakis, Dominik Pajak, Najmeh Taleb. Patrolling by Robots Equipped with Visibility. SIROCCO - 21th International Colloquium on Structural Information and Communication Complexity, Jul 2014, Hida Takayama, Japan. pp.224-234, 10.1007/978-3-319-09620-9\_18 . hal-00996773

**HAL Id: hal-00996773**

**<https://inria.hal.science/hal-00996773>**

Submitted on 26 May 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Patrolling by Robots Equipped with Visibility <sup>\*</sup>

Jurek Czyzowicz <sup>†</sup>    Evangelos Kranakis <sup>‡</sup>    Dominik Pajak <sup>§</sup>    Najmeh Taleb <sup>†</sup>

May 26, 2014

## Abstract

We study the problem of mobile robots with distinct visibility ranges patrolling a curve. Assume a set of  $k$  mobile robots (patrolmen)  $a_1, a_2, \dots, a_k$  walking along a unit-length curve in any of the two directions, not exceeding their maximal speeds. Every robot  $a_i$  has a range of visibility  $r_i$ , representing the distance from its current position at which the robot can see in each direction along the curve. The goal of the patrolling problem is to find the perpetual movement of the robots minimizing the maximal time when a point of the curve remains unseen by any robot.

We give the optimal patrolling algorithms for the case of close curve environment (known as the boundary patrolling problem in the robotics literature) and open curve (fence patrolling), when all robots have the same maximal speed. We briefly discuss the case of distinct speeds, showing that the boundary patrolling problem for robots with distinct visibility ranges is essentially different than the case of point visibility robots. We also give the optimal algorithm for fence patrolling by two robots with distinct speeds and visibility ranges.

For the case when the environment in which the robots operate is a general graph, we show that the patrolling problem for robots with distinct visibility ranges is NP-hard, while it is known that the same problem for point-visibility robots has been known to have a polynomial-time solution.

## 1 Introduction

A set of  $k$  mobile robots  $a_1, a_2, \dots, a_k$ , each one able to observe some neighborhood of its current position, has to protect (*patrol*) a given region. For this purpose the robots move perpetually around the region in order to see each point of their environment as often as possible. In this paper we study robots, moving with speeds not exceeding a certain maximal velocity, inside a uni-dimensional region represented by a unit segment or a unit-length cycle. The objective of this paper is to design algorithms producing the movements of the robots which minimize the time interval when some points of the environment remain unseen by all robots, taken over all points of the given domain being patrolled.

---

<sup>\*</sup>This work was partially supported by NSERC grants

<sup>†</sup>Jurek.Czyzowicz@uqo.ca, Département d'informatique, Université du Québec en Outaouais, Gatineau, Canada.

<sup>‡</sup>{kranakis,najmehtaleb}@scs.carleton.ca, School of Computer Science, Carleton University, Ottawa, Canada.

<sup>§</sup>dominik.pajak@inria.fr, Inria Bordeaux Sud-Ouest, 351 cours de la Liberation, 33400 Talence, France.

## 1.1 Preliminaries and Notation

Each robot  $a_i$  is equipped with visibility allowing it to see its environment within its visibility radius  $r_i$  in both directions from its current position. The visibility ranges of all robots may be different. During the movement of the set of robots, at each time  $t$  a point  $p$  of the environment is called *protected* if it is seen by at least one robot. In other words, if robot  $a_i$  protects point  $p$  at time  $t$  the distance of the robot  $a_i$  from  $p$  at time  $t$  must be at most equal to  $r_i$ . Given a perpetual movement of all robots produced by some patrolling algorithm, by the *idle time* of point  $p$  we mean the smallest value  $I(p)$ , such that in every time interval  $[t, t + I(p)]$  point  $p$  is protected by some robot. By the idle time of such an algorithm we mean the maximal value of  $I(p)$  taken over all points of the environment. Throughout the paper we will assume that  $\sum_{i=1}^k 2r_i < 1$ , since otherwise agents could constantly observe the environment without need to move.

## 1.2 Related work

Patrolling has been intensely studied in the last decade by the robotics community (cf. [3, 17, 18]). It is defined as the act of monitoring consisting in traveling around an environment in order to protect or supervise it. Patrolling is useful, e.g., to identify humans or objects of interest that need to be rescued from a disaster. Network administrators use patrolling by mobile robots to determine web pages which must be indexed by search engines or to detect network deficiencies.

Earlier work on patrolling was mainly experimental and studying heuristic methods (cf. [14, 12] though [6] brings up a theoretical analysis of the methods of patrolling. The two basic strategies to patrolling discussed in [6] are *cyclic strategy* and the *partition strategy*. In a cyclic approach, a cycle inside the environment is identified and the robots walk around this cycle in the same direction. In the partition approach the environment is divided into subregions (which may be sometimes overlapping) that are assigned to different robots. [6] was first to introduce the notion of *idleness*, which has been most often used to measure the performance of patrolling. Several other issues related to the patrolling problem were also studied, e.g. coordination and cooperation of multi-agent teams ([2, 15, 16]), dynamically changing environments or robot teams ([18, 19]) dealing with adversarial environments ([1, 4]) and many others.

Recently several interesting algorithmic issues related to patrolling were investigated. The ant-like mobile agents were used by [20] to realize an interesting distributed strategy attaining patrolling by agents traversing an Eulerian cycle of an input graph, while [10] used ant-like agents to partition the graph to patrol among them.

The optimality of the fundamental partition strategy (in the case of fence patrolling) and the cyclic strategy (for boundary patrolling) has been proven for robot teams having the same maximal speed and for small sets of robots with distinct maximal speeds (cf. [8, 13]). However, for distinct-speed sets of robots, both these strategies have been proven sub-optimal for boundary patrolling by at least three robots, [8], and fence patrolling by at least six robots, [13]. [5] presented examples with several other non-standard strategies for fence and boundary patrolling.

Same-speed robots were used in [7] for fence and boundary patrolling where some *neutral* regions may be left unprotected and in [9], where an optimal patrolling algorithm for graphs is proposed.

[13] also considered *weighted* patrolmen, i.e. such that any point  $p$  could be left unvisited for a time equal to the weight of the robot which was last to visit  $p$ . To the best of our knowledge, patrolmen equipped with visibility have not studied in the scientific literature before.

### 1.3 Outline and results of the paper

In Section 2 we show that a version of the cyclic strategy is optimal for boundary patrolling. In Section 3 we prove that the partition strategy achieves the optimal idle time for fence patrolling. To this end we show that any patrolling algorithm using robots equipped with visibility may be converted to a strategy for robots without visibility achieving at least the same idle time on a fence shortened by twice the sum of all robots' visibility radii. This could suggest that the patrolling problem for robots with visibility is equivalent to patrolling with zero-visibility robots considered elsewhere. However the hardness of the problem for general graphs shown in Section 4, in view of the polynomial solution from [9] contradicts this supposition. In Sections 2.2 and 3.2 we discuss the cases of two robots with distinct speeds and visibility radii. In Section 3.2 we show that it is possible to extend the proof of optimality of the partition strategy (cf. [8, 13]) on the case of two visibility-equipped, distinct-speed robots. However the example from Section 2.2 shows that for the circle patrolling with two distinct-speed robots the cyclic strategy from [8] is no longer optimal when the robots are equipped with visibility. This is another evidence that patrolling with visibility-equipped robots presents new challenges, even for the case of robots with the same maximal speed.

## 2 Circle patrolling

In this section we investigate patrolling of a circle. First we give the optimal patrolling for the case of any number of robots with identical speeds. In the second subsection we give the optimal algorithm for the case of two robots with distinct speeds. The case of three or more robots remains open.

### 2.1 Equal speeds

We start by considering the case of agents with equal speeds. We will assume that the maximum speed of each agent is equal to 1. Recall that 1 is also the length of the environment i.e., we consider unit circles and unit intervals.

---

**Algorithm  $A_1$**  [for  $k$  robots with the same speed and different visibility to patrol a circle]

---

1. If  $\sum_{i=1}^k 2r_i < 1$ , place the robots  $a_1, a_2, \dots, a_k$  such that the distance between robots  $i$  and  $i+1$  is equal to  $\frac{1 - \sum_{i=1}^k 2r_i}{k} + r_i + r_{i+1}$  around the circle in the counterclockwise direction and distance between robots  $k$  and 1 is  $\frac{1 - \sum_{i=1}^k 2r_i}{k} + r_k + r_1$  in the counterclockwise direction.
  2. For each  $i = 1, \dots, k$  robot  $a_i$  moves perpetually counterclockwise around the circle at maximum possible speed 1.
- 

**Proposition 1.** *Algorithm  $A_1$  achieves the idle time  $T = \frac{1 - \sum_{i=1}^k 2r_i}{k}$ .*

*Proof.* We need to show that for any point  $p$ , the longest contiguous interval of time when  $p$  is not observed by any robot has length at most  $\frac{1 - \sum_{i=1}^k 2r_i}{k}$ . Observe that all robots are moving in the same direction with the same speed 1. Thus the distances between robots and order of robots does

not change. Moreover we can observe from the definition of the algorithm that distances between extremal points visible by neighboring robots is equal to  $\frac{1-\sum_{i=1}^k 2r_i}{k}$  and is also constant in time.

Assume that point  $p$  is not observed in the interval  $J = [t_1, t_2]$ . Time  $t_1$  is the last moment of time when  $p$  is observed by a robot  $i$  and point  $t_2$  is a first moment when  $p$  is observed by a robot  $i+1$ . Thus during time interval  $J$  point  $p$  is between the regions of visibilities of robots  $i$  and  $i+1$ . By the definition of the algorithm distance between extremal points visible by robot  $i$  and  $i+1$  is constant and equal to  $\frac{1-\sum_{i=1}^k 2r_i}{k}$ . Thus the length of the interval  $J$  is exactly equal to  $\frac{1-\sum_{i=1}^k 2r_i}{k}$  which implies that the idle time of algorithm  $A_1$  equals to  $\frac{1-\sum_{i=1}^k 2r_i}{k}$ . ■

**Theorem 2.** *Any patrolling algorithm for  $k$  robots with speeds  $v_1, \dots, v_k$  and visibilities  $r_1, \dots, r_k$ , patrolling unit circle achieves idle time at least  $I_{opt} \geq \frac{1-\sum_{i=1}^k 2r_i}{\sum_{i=1}^k v_i}$ .*

*Proof.* Consider any algorithm  $A$  and its idle time  $I_A$ . Take any moment of time  $t$ . Regardless of the positions of the robots, the total length of the subset of the circle being within the radius of visibility of some robot is at most  $\sum_{i=1}^k 2r_i$ . Thus the total length of points not being observed at time  $t$  is  $1 - \sum_{i=1}^k 2r_i$ . Denote this set of not observed points at time  $t$  by  $U$ . Take the interval of time

$$J = \left[ t, t + \frac{1 - \sum_{i=1}^k 2r_i}{\sum_{i=1}^k v_i} - \epsilon \right],$$

for any  $\epsilon > 0$ . The interval has length  $\frac{1-\sum_{i=1}^k 2r_i}{\sum_{i=1}^k v_i} - \epsilon$  thus the set of all points from set  $U$  patrolled by the robot  $i$  within interval  $J$  has length at most  $v_i \frac{1-\sum_{i=1}^k 2r_i}{\sum_{i=1}^k v_i} - v_i \epsilon$ . Thus the set of all points from set  $U$  patrolled by all robots within interval  $J$  has total length at most

$$\sum_{i=1}^k v_i \frac{1 - \sum_{i=1}^k 2r_i}{\sum_{i=1}^k v_i} - \sum_{i=1}^k v_i \epsilon = 1 - \sum_{i=1}^k 2r_i - \sum_{i=1}^k v_i \epsilon = |U| - \sum_{i=1}^k v_i \epsilon < |U|.$$

Since within interval  $J$  robots are unable to patrol all points from set  $U$  thus the idle time of algorithm  $A$  is bounded from below by the length of the interval  $J$

$$I_A \geq |J| = \frac{1 - \sum_{i=1}^k 2r_i}{\sum_{i=1}^k v_i} - \epsilon.$$

Therefore the claim of the theorem is obtained by passing to the limit  $\epsilon \rightarrow 0$ . ■

**Corollary 3.** *Algorithm  $A_1$  achieves an optimal idle time in the case of robots with equal speeds and possibly different visibilities.*

## 2.2 Different speeds

In the case of equal speeds our results are the same as for the problem without visibility on a circle of length  $1 - \sum_{i=1}^k 2r_i$ . It turns out that this is not the case any more for the case of different speeds. Consider the optimal algorithm for two robots without visibilities with speeds  $v_1 > v_2$ . It is either an algorithm where both robots are at antipodal positions and move with the slower speed  $v_2$  or it is an algorithm where the faster robot goes around the circle with his maximum speed  $v_1$  and the

movement of the slower robot irrelevant. It was proven in [8] that the idle time of such algorithm is optimal. But in our problem in the case of different visibilities in some cases neither of these algorithms is optimal. In particular, when one robot is very fast with small visibility radius, while the other robot is slow but it has a large visibility radius, the partition strategy, when both robots zigzag, protecting two interior-disjoint segments of the circle, such strategy may give a better idle time bound than those obtained by the two algorithms mentioned above. It is easy to verify that this is the case for  $v_1 = 5$ ,  $r_1 = 1/12$ ,  $v_2 = 1$  and  $r_2 = 1/3$ .

### 3 Segment patrolling

In this section we investigate patrolling of a segment. First we give the optimal patrolling for the case of any number of robots with identical speeds. In the second subsection we give the optimal algorithm for the case of two robots with distinct speeds.

#### 3.1 Equal speeds

---

**Algorithm  $A_2$**  [for  $k$  robots with the same speed and different visibility to patrol a segment]

1. If  $\sum_{i=1}^k 2r_i < 1$ , partition the unit segment into  $k$  segments, such that the length of the  $i$ -th segment  $s_i$  equals  $\frac{1 - \sum_{i=1}^k 2r_i}{k} + 2r_i$ .
  2. For each  $i = 1, \dots, k$  place robot  $a_i$  at the center of the segment  $s_i$ .
  3. For each  $i = 1, \dots, k$  robot  $a_i$  moves perpetually at maximal speed and changes its direction when being at distance  $r_i$  from an endpoint of  $s_i$ .
- 

**Proposition 4.** *If  $\sum_{i=1}^k 2r_i < 1$  then patrolling algorithm  $A_2$  achieves idle time  $T = 2 \frac{1 - \sum_{i=1}^k 2r_i}{k}$ .*

*Proof.* Since each robot covers a non-overlapping sub-segment of the unit segment (except for the common endpoint, which may be visited by two robots) the interior points of each sub-segment  $s_i$  are visited by the same robot  $a_i$ . The infimum of the frequency of visits of point  $x$  inside  $s_i$  is achieved for  $x$  being its endpoint. Since between two consecutive visits to the endpoint  $x$  of  $s_i$  robot  $a_i$  traverses, using its speed  $v_i$ , the sub-segment  $s_i$  of length  $\frac{1 - \sum_{i=1}^k 2r_i}{k}$  twice, the idle time of such a point is  $T = 2 \frac{1 - \sum_{i=1}^k 2r_i}{k}$ . ■

We want to prove optimality of the algorithm  $A_2$ . For any patrolling algorithm  $A^{vis}$  with visibility, operating on a segment of length 1 we will construct an algorithm  $A^{vis}$  with no visibility, working on a segment of length  $1 - \sum_{i=1}^k 2r_i$ . The construction will ensure that the idle time of  $A^{vis}$  is bigger than or equal to the idle time of  $A^{vis}$ . Then, since it is straightforward to show the optimal algorithm for robots with the same speed and no visibility, we will obtain a desired lower bound.

Take any algorithm  $A^{vis}$  for  $k$  robots with different visibilities  $r_1, r_2, \dots, r_k$  and the same speed 1 working on the line segment of length 1. Assume that  $\sum_{i=1}^k 2r_i < 1$ . Denote by  $A_i^{vis}(t)$  the position of robot  $i$  in time  $t$ . Define functions  $L_i(t) = A_i^{vis}(t) - r_i$  and  $R_i(t) = A_i^{vis}(t) + r_i$  for any

robot  $i$  and time  $t$ . Function  $L_i(t)$  denotes the leftmost point visible to robot  $i$  in time  $t$ , except the case when robot  $i$  is too close to the left endpoint 0. Similarly, function  $R_i(t)$  denotes the rightmost point visible to robot  $i$  in time  $t$ , except the case when robot  $i$  is too close to the right endpoint 1.

**Definition of mobile intervals.** A mobile interval is a pair of functions  $\langle L(t), R(t) \rangle$  denoting its left and right endpoints respectively. The first step of our construction is the definition of a set of mobile intervals on the line. The goal of the construction is to obtain a dynamic process such that at any moment of time the union of all mobile intervals covers all points that are visible to all robots. We will require that mobile intervals move with speed not exceeding 1 and total length of the intervals is always equal to  $\sum_{i=1}^k 2r_i$ . If at some time  $t$  for all  $i \in \{1, 2, \dots, k\}$ , the length of the set of points visible to a robot  $i$  is equal to  $2r_i$ , (no robot is too close to the endpoint) and all sets of visible points are disjoint (the visibility regions of two robots are always interior-disjoint), then mobile intervals will simply have  $k$  intervals with lengths equal to areas of visibilities. However, for such time moments  $t$  when the areas of visibilities of the robots are overlapping then the mobile intervals at time  $t$  will cover some points that are not being observed by any robot. Now we will present an algorithm that determines the positions of mobile intervals depending on the positions of robots in the algorithm  $A^{vis}$  at any time moment  $t$ .

---

**Procedure 1** Construction of mobile intervals

---

```

1:  $J \leftarrow \{\langle L_i(t), R_i(t), 1 \rangle, i \in 1, 2, \dots, k\}$   $\triangleright$  initially intervals are equal to areas of visibilities
2: while there exists  $\langle L, R, h \rangle \in J$  such that  $L < 0$  do  $\triangleright$  if left endpoint is beyond the point 0
3:    $J \leftarrow J \setminus \{\langle L, R, h \rangle\} \cup \{\langle 0, R - L, h \rangle\}$   $\triangleright$  move the interval to the right
4: end while
5: while there exists  $\langle L, R, h \rangle \in J$  such that  $R > 1$  do  $\triangleright$  if right endpoint is beyond the point 1
6:    $J \leftarrow J \setminus \{\langle L, R, h \rangle\} \cup \{\langle 1 - (R - L), 1, h \rangle\}$   $\triangleright$  move the interval to the left
7: end while
8: while there exist  $\langle L, R, h \rangle, \langle L', R', h' \rangle \in J$  such that  $L < L' \leq R$  do  $\triangleright$  overlapping intervals
9:    $j \leftarrow \langle L, R + R' - L', h + h' \rangle$   $\triangleright$  merge the pair into one long interval
10:  if  $R + R' - L' > 1$  then  $\triangleright$  if the new interval would go beyond the endpoint
11:     $j \leftarrow \langle 1 - (R - L) - (R' - L'), 1, h + h' \rangle$   $\triangleright$  move it to the left
12:  end if
13:   $J \leftarrow (J \setminus \{\langle L, R, h \rangle, \langle L', R', h' \rangle\}) \cup \{j\}$   $\triangleright$  replace the pair with newly constructed interval
14: end while
15:  $J(t) \leftarrow \{j_1, j_2, \dots, j_l\} \leftarrow$  sort set  $J$  according to left endpoints of the intervals
16: return  $J(t)$ 

```

---

If  $t$  is the real variable denoting time then  $J(t)$  is a dynamic process in which the mobile intervals are moving on the interval  $[0, 1]$ . In the following lemma we prove some properties of this process.

**Lemma 5.** *At any time moment  $t$ , the sequence of intervals  $J(t)$  returned by the Procedure 1 satisfy*

- (1) *the intervals from  $J(t)$  cover all points visible to the robots in the algorithm  $A^{vis}$  at time  $t$ ,*
- (2) *the intervals from  $J(t)$  are interior-disjoint and have total length equal to  $\sum_{i=1}^k 2r_i$ ,*
- (3) *the velocity of any interval  $j \in J(t)$  is either equal to velocity of some robot  $a$  or it is equal to 0.*

*Proof.* Observe that the output  $J(t)$  of the Procedure 1 is a sequence of intervals that are disjoint because the while loop in lines 8–14 is executed until there are some interleaving intervals. Moreover initially (line 1) the intervals cover all points that are visible to all robots and any operation in the Procedure 1 preserves this property. Thus the output  $J(t)$  is a sequence of intervals that cover all points that are visible to robots in algorithm  $A^{vis}$ . And since the intervals are interior-disjoint they have total length  $\sum_{i=1}^k 2r_i$ . To prove (3) let  $L$  be the left endpoint of interval  $j$ . We can find  $\delta_t > 0$  such that for any  $t' \in [t, t + \delta_t]$ , the Procedure 1 performs exactly the same operations on intervals. Consider two cases. Either  $L$  is equal to the left endpoint of the area of visibility of some robot  $a$  (i.e.  $L = L_a(t)$ ). In this case the left endpoint of the interval  $j$  for any  $t' \in [t, t + \delta_t]$  is equal to the left endpoint of the area of visibility of robot  $a$ . Thus the velocity of interval  $j$  in time  $t$  is equal to the velocity of robot  $a$ . In the other case the position of  $j$  for any  $t' \in [t, t + \delta_t]$  is constant thus the velocity of  $j$  is equal to 0. ■

Thus any mobile interval either moves with the same speed and direction as some robot, or merges with other interval or splits into multiple intervals.

Based on the positions of the intervals we can define positions of the robots in algorithm  $A^{vis}$  at any step  $t$ .

**Definition of the algorithm  $A^{vis}$ .** Take any time moment  $t$ . Based on the positions of robots in the algorithm  $A^{vis}$  at this moment we construct the mobile intervals  $J(t)$ . Consider the output of the Procedure 1, namely the set  $J(t) = \{j_1, j_2, \dots, j_l\}$ . Each  $j_i$  is a tuple  $j_i = \langle L_i, R_i, h_i \rangle$ , where  $L_i$  is the left endpoint,  $R_i$  is the right endpoint and  $h_i$  is the number of robots whose areas of visibilities are being covered by interval  $j_i$ . Recall that  $J(t)$  is sorted thus  $L_1 < R_1 < L_2 < R_2 < \dots < L_l < R_l$ . To obtain positions of the robots in  $A^{vis}$  we intuitively cut the terrain that is covered by the mobile intervals (see Figure 1). The amount of terrain that is left is  $1 - \sum_{i=1}^k 2r_i$ . Define points  $p_1 = L_1$ ,

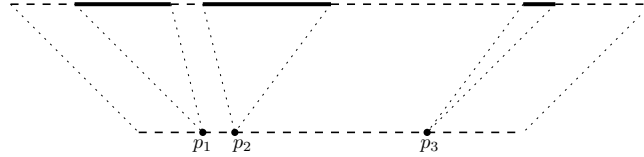


Figure 1: Positions of robots in the algorithm  $A^{vis}$ .

$p_i = L_i - \sum_{j=1}^{i-1} (R_j - L_j)$  for  $i = 2, 3, \dots, l$  as the total length of all points to the left of point  $L_i$  not covered by any mobile interval. Points  $p_i$  will be positions of the robots in the algorithm  $A^{vis}$ . Number of robots located at the point  $p_i$  will be equal to  $h_i$ . Let  $s_i = \sum_{j=1}^i h_j$  for  $i = 1, 2, \dots, l$ . Positions of robots in algorithm  $A^{vis}$  are defined as follows.

$$\begin{aligned} A_a^{vis}(t) &= p_1 && \text{for } a = 1, 2, \dots, s_1 \\ A_a^{vis}(t) &= p_2 && \text{for } a = s_1 + 1, s_1 + 2, \dots, s_2 \\ &\vdots \\ A_a^{vis}(t) &= p_l && \text{for } a = s_{l-1} + 1, s_{l-1} + 2, \dots, s_l \end{aligned}$$



This completes the construction of algorithm  $A^{vis}$ . We want to prove that algorithm  $A^{vis}$  has idle time not bigger than  $A^{vis}$ . First we need to show that robots in the algorithm  $A^{vis}$  move with speeds not exceeding 1.

**Lemma 6.** *For any time moment  $t$  and robot  $a_i$*

- (1)  $A_i^{vis}(t) \in [0, 1 - \sum_{i=1}^k 2r_i]$ ,
- (2) *speed of robot  $a_i$  is at most 1,*
- (3) *the trajectory followed by robot  $a_i$  is continuous.*

*Proof.* First we want to prove part (1). Recall that the positions of robots in algorithm  $A^{vis}$  at time  $t$  are defined as points  $p_i$  satisfying  $p_i = L_i - \sum_{j=1}^{i-1} (R_j - L_j)$ , for  $i = 1, 2, \dots, l$ , where  $R_i$  and  $L_i$  are positions of the endpoints of the mobile intervals at time  $t$ . Intervals are ordered (line 15 of the Procedure 1) and non-overlapping (Lemma 5(2)) thus  $L_1 < R_1 < L_2 < R_2 < \dots < L_l < R_l$ . Observe that  $p_{i+1} - p_i = L_{i+1} - R_i > 0$ , thus the sequence  $p_1, p_2, \dots, p_l$  is increasing. Since  $p_1 \geq 0$  we have  $A_i^{vis}(t) \geq 0$ . Then

$$\begin{aligned} p_l &= L_l - \sum_{j=1}^{l-1} (R_j - L_j) \\ &= R_l - \sum_{j=1}^l (R_j - L_j) \\ &\leq 1 - \sum_{i=1}^k 2r_i, \end{aligned}$$

where in the last inequality we use the fact that the total length of the mobile intervals is equal to  $\sum_{i=1}^k 2r_i$  (which is true by Lemma 5(2)). Since  $p_l$  is the largest value  $p_i$  we have that  $A_i^{vis}(t) \leq 1 - \sum_{i=1}^k 2r_i$ .

The proof of part (2) is a consequence of Lemma 5(3).

It remains to prove part (3). Observe that movement of mobile intervals is continuous thus movement of the positions  $p_i$  is also continuous. It is left to observe that a robot may not "jump" between positions  $p_i$ . Indeed, it is true because the number of robots assigned to mobile intervals (variables  $h_j$  in the Procedure 1) changes only if mobile intervals join or split. ■

Lemma 6 shows that algorithm  $A^{vis}$  is a correct patrolling algorithm for segment of length  $1 - \sum_{i=1}^k 2r_i$  and robots with the same speed 1. Now we need to show that the transformation does not increase the idle time.

**Lemma 7.** *The idle time of algorithm  $A^{vis}$  is not larger than the idle time of algorithm  $A^{vis}$ .*

*Proof.* Let  $I^{vis}$  and  $I^{vis}$  be idle times of algorithms  $A^{vis}$  and  $A^{vis}$  respectively. Consider the parallel execution of both algorithms  $A^{vis}$  and  $A^{vis}$ . Take a vertex  $v \in [0, 1 - \sum_{i=1}^k 2r_i]$  and an interval of time  $[t_1, t_2]$  and assume that  $v$  is not visited in algorithm  $A^{vis}$  in interval  $[t_1, t_2]$ . Fix a vertex  $v'$  such that in time  $t_1$  total length of the regions not covered by the mobile intervals on the left of  $v'$  is equal to  $v$ , and  $v'$  is a leftmost point with such property. Since  $v'$  is leftmost then it

is either a left endpoint of some mobile interval or it is not covered by any mobile interval at time  $t_1$ . If  $v'$  was a left endpoint of some mobile interval then by the definition of algorithm  $A^{vis}$ , the corresponding point  $v$  would be visited at time  $t_1$  by algorithm  $A^{vis}$ . Thus  $v'$  is not covered by any mobile interval at time  $t_1$ . Using the same argument we can prove that  $v'$  is not covered by any mobile interval at any moment from time interval  $[t_1, t_2]$ . Since mobile intervals at any time cover all points visible to robots in  $A^{vis}$  then  $v'$  is also not refreshed by any robot in algorithm  $A^{vis}$ . Now if  $I^{vis} > I^{vis}$  then there exists a vertex  $v^*$  and a closed interval  $[t^*, t^{**}]$  of length larger than  $I^{vis}$  such that algorithm  $A^{vis}$  does not visit  $v^*$  in this interval. But this would imply existence of a point not refreshed by  $A^{vis}$  in the interval  $[t^*, t^{**}]$  which is a contradiction since length of  $[t^*, t^{**}]$  is larger than  $I^{vis}$ . ■

**Theorem 8.** *The optimal traversal algorithm for  $k$  robots with the same speed and different visibility, patrolling unit segment  $S = [0, 1]$  achieves idle time  $I_{opt} = 2 \frac{1 - \sum_{i=1}^k 2r_i}{k}$ .*

*Proof.* By our construction and Lemma 7 existence of an algorithm  $B^{vis}$  for robots with visibility operating on a segment  $[0, 1]$  achieving idle time  $I$  implies existence of an algorithm  $B^{vis}$  for robots with no visibility operating on a segment  $[0, 1 - \sum_{i=1}^k 2r_i]$  with idle time at most  $I$ . The optimal algorithm for patrolling without visibility of a segment of length  $1 - \sum_{i=1}^k 2r_i$  achieves idle time  $2 \frac{1 - \sum_{i=1}^k 2r_i}{k}$  [13]. Thus there cannot exist an algorithm with visibility achieving idle time better than  $2 \frac{1 - \sum_{i=1}^k 2r_i}{k}$  on the unit segment. Hence  $2 \frac{1 - \sum_{i=1}^k 2r_i}{k}$  is a lower bound on idle time for algorithms with visibility on the unit segment. By Proposition 4 algorithm  $A_2$  achieves the optimal idle time. ■

### 3.2 Different speeds

---

**Algorithm  $A_3$**  [for 2 robots with different speed and visibility to patrol a segment]

---

1. Partition the unit segment into 2 segments, such that the length of the  $i$ -th segment  $s_i$  equals  $\frac{v_i(1-2(r_1+r_2))}{v_1+v_2}$ .
  2. For each  $i = 1, 2$  place robot  $a_i$  at the center of the segment  $s_i$ .
  3. For each  $i = 1, 2$  robot  $a_i$  moves perpetually at maximal speed and changes its direction when being at distance  $r_i$  from an endpoint of  $s_i$ .
- 

**Proposition 9.** *Traversal algorithm  $A_3$  achieves idle time  $T = 2 \frac{1-2(r_1+r_2)}{v_1+v_2}$ .*

*Proof.* Since each robot covers a non-overlapping sub-segment of the unit segment (except for the common endpoint, which may be visited by two robots) the interior points of each sub-segment  $s_i$  are visited by the same robot  $a_i$ . The infimum of the frequency of visits of point  $x$  inside  $s_i$  is achieved for  $x$  being its right endpoint. Take a point  $x - \epsilon$ , for  $\epsilon > 0$ . Since between two consecutive visits to the endpoint  $x$  of  $s_i$  robot  $a_i$  traverses, using its speed  $v_i$ , the sub-segment  $s_i$  of length  $\frac{v_i(1-2(r_1+r_2))}{v_1+v_2} - \epsilon$  twice, the idle time of such a point is  $T = 2 \frac{1-2(r_1+r_2)}{v_1+v_2} - \epsilon$ . As  $\epsilon$  may be as close to 0 as we wish, we obtain the claim of our lemma. ■

We prove below that the algorithm  $A_3$  is optimal.

**Theorem 10.** *The optimal traversal algorithm for two robots with different speed and visibility, patrolling unit segment  $S = [0, 1]$  achieves idle time  $I_{opt} = 2 \frac{1-2(r_1+r_2)}{v_1+v_2}$ .*

*Proof.* We suppose, by contradiction, that there exists an algorithm  $A$  with an idle time of  $I_A = T - \epsilon$  for some  $\epsilon > 0$ . Without loss of generality, we may assume that  $v_1 \leq v_2$ . Observe that  $a_1$  must see one of the endpoints  $(0, 1)$ . By symmetry suppose that  $a_1$  sees endpoint 0 at some time  $t_1$ . Let  $L_1 = r_1$ ,  $R_1 = r_1 + \frac{1-2(r_1+r_2)}{v_1+v_2}v_1$ ,  $L_2 = 1 - r_2 - \frac{1-2(r_1+r_2)}{v_1+v_2}v_2$ ,  $R_2 = 1 - r_2$ , and  $B = R_1 + r_1 = L_2 - r_2$ . At time  $t_1$ ,  $a_1$  is within  $[0, L_1]$ . Considering the speed of  $a_1$ ,

$$v_1 \frac{T - \epsilon}{2} + r_1 < |B - 0|$$

and  $a_1$  cannot see  $B$  within time  $[t_1 - \frac{T-\epsilon}{2}, t_1 + \frac{T-\epsilon}{2}]$ . So,  $a_2$  has to see  $B$  at some time  $t_2$ . We will show that neither  $a_2$  nor  $a_1$  can see 1 within  $[t_2 - \frac{T-\epsilon}{2}, t_2 + \frac{T-\epsilon}{2}]$ . Considering the speed of  $a_2$ ,  $v_2 \frac{T-\epsilon}{2} + r_2 < |B - 1|$  and  $a_2$  cannot see 1 within the time interval  $[t_2 - \frac{T-\epsilon}{2}, t_2 + \frac{T-\epsilon}{2}]$ . Now, we show that  $a_1$  cannot see 1 neither. Since  $|t_1 - t_2| < \frac{T}{2}$ , the rightmost point  $p$  at which  $a_1$  can be at time  $t_2$  is  $p < L_1 + \frac{T}{2}v_1$ . We show that from the moment when  $a_1$  sees point  $p$ , it is not possible for  $a_1$  to see 1 within time  $\frac{T-\epsilon}{2}$ , this would prove that in interval  $[t_2 - \frac{T-\epsilon}{2}, t_2 + \frac{T-\epsilon}{2}]$ ,  $a_1$  cannot see 1. The rightmost point  $q$  that  $a_1$  can visit within time  $t_2 + \frac{T-\epsilon}{2}$  is  $q < p + \frac{T-\epsilon}{2}v_1$ . Hence, the rightmost point that  $a_1$  can see within time  $t_2 + \frac{T-\epsilon}{2}$  is

$$q + r_1 < L_1 + \frac{T}{2}v_1 + \frac{T-\epsilon}{2}v_1 = L_1 + Tv_1 - \frac{\epsilon}{2}v_1,$$

which proves the theorem. ■

## 4 Hardness results

Let us recall the definition of the PARTITION problem.

**Instance:** Finite set  $A$  and size  $s(a) \in \mathbb{Z}^+$  for each  $a \in A$ .

**Question:** Is there a subset  $A' \subset A$  such that  $\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)$ ?

The PARTITION problem remains NP-complete even if we require that  $|A'| = |A|/2$  [11]. We will refer to the PARTITION problem with this additional condition as RESTRICTED PARTITION problem.

**Theorem 11.** *For some graphs the problem of deciding, for any set of robots  $a_1, a_2, \dots, a_k$  with equal speeds and different visibilities, whether there exists a patrolling algorithm with idle time 0 is NP-hard.*

*Proof.* We want to reduce the partition problem, which is known to be NP-hard. Assume that there exists an algorithm  $A$  answering, given a graph and a set of robots, whether it is possible to deploy the robots in the graph such that their visibility ranges cover the entire graph (i.e. the idle time is 0). Let a multiset of integers  $S = \{x_1, x_2, \dots, x_n\}$  be an instance from the PARTITION problem. Let  $s = \sum_{i=1}^n x_i$ . We will construct an instance of the patrolling problem as follows. Take  $n + 1$

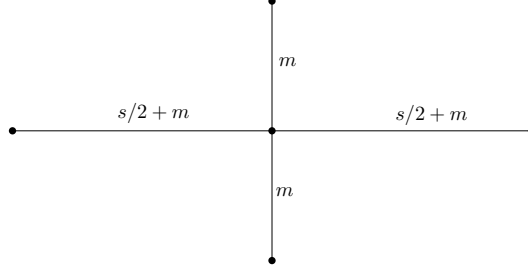


Figure 2: Graph  $G$ .

robots  $a_1, a_2, \dots, a_{n+1}$  and let radius of visibility of  $i$ -th robot be equal to  $x_i/2$  for  $i = 1, 2, \dots, n$  and let radius of visibility of  $n + 1$ -st robot be equal to  $m = \max\{x_1, x_2, \dots, x_n\} + 1$ . Let graph  $G$  be a "cross" with two arms of length  $m$  and two of length  $s/2 + m$  (see Figure 2). A patrolling algorithm with idle time 0 is simply a placement of robots such that every point is observed (i.e. within radius of visibility) for some robot. The horizontal line of the cross  $G$  has length  $s + 2m$  which is equal to sum of diameters of visibilities of all robots. Thus any algorithm which places any robot not on the horizontal line will not have idle time 0. Moreover to cover the vertical line an algorithm has to place the  $n + 1$ -st robot on the crossing. To cover the remaining part of  $G$  an algorithm has to divide the robots into two groups with equal total length of visibility  $s/2$ . Thus  $G$  can be patrolled with idle time 0 if and only if there is a solution to the partition problem. Thus having algorithm  $A$  we can construct an algorithm solving the partition problem. ■

When the idle time is strictly positive the construction of the example showing NP-hardness is more involved.

**Theorem 12.** *For any fixed  $I$  and for some graphs the problem of deciding, for any set of robots  $a_1, a_2, \dots, a_k$  with equal speeds and different visibilities, whether there exists a patrolling algorithm with idle time at most  $I$  is NP-hard.*

*Proof.* Fix any  $I$ . Assume that there exists a polynomial algorithm deciding for any set of robots whether it is possible to patrol a graph obtaining idle time at most  $I$ . We want to show that it would imply existence of a polynomial algorithm for the RESTRICTED PARTITION problem.

Let a multiset of integers  $S = \{x_1, x_2, \dots, x_{2k}\}$  be an instance from the RESTRICTED PARTITION problem. We construct an instance of patrolling problem consisting of  $2k + 1$  robots in the following way. Let the radius of visibility  $r_i$  of  $i$ -th robot be  $r_i = I/4 \sum_{j=1}^{2k} x_j$  for  $i = 1, 2, \dots, 2k$  and let the radius of visibility of  $(2k + 1)$ -st robot (call it  $a^*$ ) be  $r_{2k+1} = I$ . We take the following graph  $H$  (see Figure 3). The length of interval  $AB$  is  $I/2$ , the length of each interval  $AA_i$  is  $I$  for  $i = 1, 2, \dots, k + 1$ , the length of each interval  $BB_i$  is  $I$  for  $i = 1, 2, \dots, k + 1$  and the radius of each circle is  $k/2 + I/4$ . We ask if such collection of robots can patrol graph  $H$  with idle time at most  $I$ .

We will argue that the answer can be yes if and only if the RESTRICTED PARTITION problem has a solution. First observe that the robot  $a^*$  has to walk perpetually between nodes  $A$  and  $B$ . Note that a robot has to visit node  $A$  at least once in every interval of time of length  $I$ . If it does not visit  $A$  in some interval of length  $I$  then in this interval some of the nodes  $A_1, A_2, \dots, A_{k+1}$  will not be patrolled by any robot. Since the distance between two nodes  $A_i, A_j$  ( $i \neq j$ ) is  $2I$  thus any robot different from  $a^*$  can patrol at most one node among  $A_1, A_2, \dots, A_{k+1}$  in time  $I$ . And since

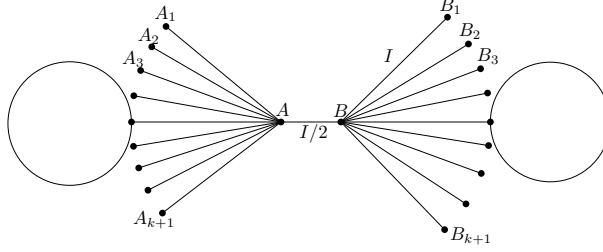


Figure 3: Graph  $H$ .

we have  $k + 1$  nodes  $A_i$  then  $a^*$  has to visit node  $A$  once every  $I$  time steps. Since the same applies to  $B$ , robot  $a^*$  has to perpetually walk between  $A$  and  $B$ . To patrol circles we have to use the remaining robots. First note that, if during some time interval of size  $I$   $k - 1$  or less robots will be on one of the circles then the idle time must be larger than  $I$ . It is because  $k - 1$  robots can patrol in time  $I$  intervals of total length at most  $(k - 1)I + I/2$  (because  $I/2$  is an upper bound on the total length of the visibilities) which is less than the length of the circle. Therefore, idle time  $I$  will be achieved if and only if the two sums of diameters of visibilities of groups assigned to both circles are the same. But this corresponds exactly to the solution of the RESTRICTED PARTITION problem. ■

## 5 Conclusion

In this paper we investigated the problem of mobile robots with visibility patrolling a curve. We gave optimal patrolling algorithms for the case of boundary patrolling and fence patrolling when all robots have the same maximal speed and discussed briefly the case of distinct speeds, thus showing that the boundary patrolling problem for robots with distinct visibility ranges is entirely different than the case of point visibility robots. We also give the optimal algorithm for fence patrolling by two robots with distinct speeds and visibility ranges. If the underlying domain in which the robots operate is a general graph, then the patrolling problem for robots with distinct visibility ranges is shown to be NP-hard; this contrasts sharply with point-visibility robots which has been known to have a polynomial-time solution [9].

There are several open problems but the most interesting class of problems seems to be related to the analysis of approximation as well as online and offline patrolling algorithms for robots with distinct visibilities and/or speeds.

## References

- [1] N. Agmon. On events in multi-robot patrol in adversarial environments. In *AAMAS*, pages 591–598, 2010.
- [2] N. Agmon, C.-L. Fok, Y. Emaliah, P. Stone, C. Julien, and S. Vishwanath. On coordination in practical multi-robot patrol. In *ICRA*, pages 650–656, 2012.
- [3] A. Almeida, G. Ramalho, H. Santana, P. A. Tedesco, T. Menezes, V. Corruble, and Y. Chevalleyre. Recent advances on multi-agent patrolling. In *SBIA*, pages 474–483, 2004.

- [4] F. Amigoni, N. Basilico, and N. Gatti. Finding the optimal strategies for robotic patrolling with adversaries in topologically-represented environments. In *ICRA*, pages 819–824, 2009.
- [5] K. Chen, A. Dumitrescu, and A. Ghosh. On fence patrolling by mobile agents. In *CCCG*, 2013.
- [6] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *IAT*, pages 302–308, 2004.
- [7] A. Collins, J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, R. Martin, and O. M. Ponce. Optimal patrolling of fragmented boundaries. In *SPAA*, pages 241–250, 2013.
- [8] J. Czyzowicz, L. Gasieniec, A. Kosowski, and E. Kranakis. Boundary patrolling by mobile agents with distinct maximal speeds. In *ESA*, pages 701–712, 2011.
- [9] J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, and D. Pajak. Optimal patrolling by mobile agents in arbitrary continuous graphs, 2014, in preparation.
- [10] Y. Elor and A. M. Bruckstein. Autonomous multi-agent cycle based patrolling. In *ANTS Conference*, pages 119–130, 2010.
- [11] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [12] N. Hazon and G. A. Kaminka. On redundancy, efficiency, and robustness in coverage for multiple robots. *Robotics and Autonomous Systems*, 56(12):1102–1114, 2008.
- [13] A. Kawamura and Y. Kobayashi. Fence patrolling by mobile agents with distinct speeds. In *ISAAC*, pages 598–608, 2012.
- [14] A. Machado, G. Ramalho, J.-D. Zucker, and A. Drogoul. Multi-agent patrolling: An empirical analysis of alternative architectures. In *MABS*, pages 155–170, 2002.
- [15] F. Pasqualetti, J. W. Durham, and F. Bullo. Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms. *IEEE Transactions on Robotics*, 28(5):1181–1188, 2012.
- [16] F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, 28(3):592–606, 2012.
- [17] D. Portugal and R. P. Rocha. A survey on multi-robot patrolling algorithms. In *DoCEIS*, pages 139–146, 2011.
- [18] D. Portugal and R. P. Rocha. Multi-robot patrolling algorithms: examining performance and scalability. *Advanced Robotics*, 27(5):325–336, 2013.
- [19] S. L. Smith, M. Schwager, and D. Rus. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Transactions on Robotics*, 28(2):410–426, 2012.
- [20] V. Yanovski, I. A. Wagner, and A. M. Bruckstein. A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37(3):165–186, 2003.